(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 July 2003 (17.07.2003)

PCT

(10) International Publication Number
WO 03/058439 A2

(51) International Patent Classification[7]: G06F 9/40

(21) International Application Number: PCT/US02/39159

(22) International Filing Date: 6 December 2002 (06.12.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/039,652 31 December 2001 (31.12.2001) US

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventors: **NALAWADI, Rajeev**; 690 Ahostini Circle, Folsom, CA 95630 (US). **SIDDIQI, Faraz**; 816 Canyon Terrace Lane, Folsom, CA 95630 (US).

(74) Agent: **MALLIE, Michael, J.**; Blakely, Sokoloff, Taylor & Zafman, 7th Floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 03/058439 A2

(54) Title: METHOD AND APPARATUS FOR MODIFYING THE CONTENTS OF A REVISION IDENTIFICATION REGISTER

(57) Abstract: An embodiment for modifying the contents of a revision identification register includes a revision identification register that is both readable and writable (the contents of the revision identification register are modifiable). A revision identification modification bit is also included. The contents of the revision identification register are only modifiable when the revision identification modification bit is set to indicate that writes to the revision identification register will be accepted.

METHOD AND APPARATUS FOR MODIFYING THE CONTENTS OF A REVISION IDENTIFICATION REGISTER

Field Of The Invention

[0001] The present invention pertains to the field of computer systems. More particularly, this invention pertains to the field of revision identification registers.

Background of the Invention

[0002] In a typical computer system, many of the computer system devices include revision identification registers that contain a value that indicates the current revision of the associated hardware. Typically, every time a device undergoes a revision (often referred to as a "stepping"), the value stored in the revision identification register is changed to reflect the new stepping. Prior computer system devices implement the revision identification registers as read-only registers. In other words, the contents of the revision identification registers are not modifiable.

[0003] When an operating system is loaded during a computer system boot process, the operating system typically checks the various revision identification registers to determine the current steppings of the various devices. The operating system uses the revision identification information to make decisions regarding which device drivers to load. When the operating system recognizes a new device stepping, the operating system loads an updated device driver for that device.

[0004] Computer systems typically include highly-integrated system logic devices often referred to as "chipsets". Theses system logic devices may include many functional units, and many of these functional units may include revision identification registers. Typically, when a highly-integrated system logic device undergoes a new stepping, most of

the functional units do not have a change in functionality to warrant a change in the device driver. However, for each new stepping, each of the revision identification registers associated with the integrated functional units contain a value that reflects the new stepping. During the boot process, if the operating system detects a new device stepping, the operating system will load updated device drivers for the various functional units even if there is no change in functionality from the previous stepping to the current stepping.

[0005] The loading of updated drivers whether needed or not when a new stepping is detected poses difficulties for computer system manufacturers. Computer system manufacturers often build a pre-load of an operating system on one system and then moves it to an identical system that may have a system logic device with a different silicon stepping. Upon power-up, the operating system will load new device drivers even if not needed, thereby increasing manufacturing time.

[0006] The revision identification registers described above are often implemented in accordance with the Peripheral Component Interconnect (PCI) bus standard (Peripheral Component Interconnect Local Bus Specification, Rev. 2.2, released December 18, 1998).

Brief Description of the Drawings

[0007] The invention will be understood more fully from the detailed description given below and from the accompanying drawings of embodiments of the invention which, however, should not be taken to limit the invention to the specific embodiments described, but are for explanation and understanding only.

[0008] Figure 1 is a block diagram of one embodiment of a computer system including modifiable revision identification registers.

[0009] Figure 2 is a flow diagram of one embodiment of a method for modifying a value stored in a revision identification register.

[0010] Figure 3 is a flow diagram of an additional embodiment of a method for modifying a value stored in a revision identification register.

5

Detailed Description

[0011] An embodiment for modifying the contents of a revision identification register includes a revision identification register that is both readable and writable (the contents of the revision identification register are modifiable). A revision identification
10 modification bit is also included. The contents of the revision identification register are only modifiable when the revision identification modification bit is set to indicate that writes to the revision identification register will be accepted. This embodiment is useful in scenarios where the hardware is updated but the operating system device drivers need not be updated for a particular functional unit or particular functional units within a highly-
15 integrated system logic device. In such cases, the pre-operating system boot software (such as the basic input/output system (BIOS) which is executed before the operating system is loaded), which is developed specifically for a particular computer system configuration, can make the decision to place an older revision identification value into the revision identification register before the operating system loads. The operating system
20 assumes that the hardware has not changed. This prevents the unnecessary enumeration, search, and reloading of the device drivers, thereby saving manufacturing time.

[0012] Figure 1 is a block diagram of one embodiment of a computer system including modifiable revision identification registers. The computer system of Figure 1

includes a processor 110. The processor 110 is coupled to a system logic device 120. The system logic device 120 may include a memory controller for communicating with a system memory 130. The system logic device 120 may also include other functional units including a graphics controller.

[0013] The system logic device 120 is also coupled to an input/output hub 140. The input/output hub 140 may include a variety of functional units, including, but not limited to, functional units that provide communication with a universal serial bus (USB) 165, a PCI bus 155, and a disk drive interface 175. The disk drive interface 175 may communicate with a storage device (not shown) that has stored thereon an operating system. All or part of the operating system may be loaded into the system memory 130 during the computer system boot process.

[0014] The input/output hub 140 is also coupled to a non-volatile memory 180. The non-volatile memory 180 may store a basic input/output system (BIOS) or other pre-operating system software. As used herein, the term "pre-operating system software" is meant to include any software agents that are executed prior to the operating system loading during the system boot process.

[0015] The computer system of Figure 1 includes a number of revision identification registers. The system logic device 120 includes a revision identification register 122 and a revision identification modification bit 124. When the revision identification modification bit 124 is set to the value "1", the revision identification register 122 will accept writes. When the revision modification bit 124 reflects a value of "0", the revision identification register 122 is read-only (will not accept writes).

[0016] The input/output hub 140 includes revision identification registers 141, 143, and 145. The input/output hub 140 also includes revision identification modification bits 142, 144, and 146. The revision identification register 141 and the revision identification modification bit 142 are associated with the functional unit that provides communication with the USB 165. The revision identification register 143 and the revision identification modification bit 144 are associated with the functional unit that provides communication with the PCI bus 155. The revision identification register 145 and the revision identification modification bit 146 are associated with the functional unit that provides communication with the disk drive interface 175. Many other possible embodiments exist where revision identification registers and revision identification modification bits are associated with any of a wide range of functional units or devices.

[0017] As with the revision identification register 122 and the revision identification modification bit 124, the revision identification registers 141, 143, and 145 are not modifiable when their associated revision identification modification bits 142, 144, and 146 contain the value "0". When the revision identification modification bits 142, 144, and 146 are set to the value "1", the associated revision identification registers 141, 143, and 145 are modifiable. That is, the revision identification registers 141, 143, and 145 will accept writes.

[0018] The pre-operating system software stored in the non-volatile memory 180 may include code that checks the current values stored in the various revision identification registers. This code can determine whether to replace the current revision identification register values with values that reflect previous device steppings.

[0019] For this example embodiment, the revision identification registers 122, 141, 143, and 145 and the revision identification modification bits 124, 142, 144, and 146 may be included as part of the PCI configuration spaces associated with the various associated devices or functional units.

[0020] Although example embodiments described herein discuss a revision identification modification bit for determining whether an associated revision identification register is modifiable, other embodiments are possible using more than one bit.

[0021] Further, although example embodiments described herein include revision identification modification bits where a value of "1" indicates that an associated revision identification register is modifiable and a value of "0" indicates a read-only state for the revision identification register, other embodiments are possible where a value of "0" indicates that an associated revision identification register is modifiable and a value of "1" indicates a read-only state for the revision identification register.

[0022] Figure 2 is a flow diagram of one embodiment of a method for modifying a value stored in a revision identification register. At block 210, a current revision identification register value is read from a revision identification register. A check is made at block 220 to determine whether the current revision identification value indicates a first device stepping. This first device stepping may be the last stepping where any significant changes in functionality occurred. The term "first device stepping" does not necessarily mean the original version of a device or functional unit.

[0023] If the current revision identification value indicates the first device stepping, then no further action is taken. However, if the current revision identification

value does not indicate the first device stepping, then at block 230 the current revision identification value is replaced with a value that indicates the first device stepping.

[0024] Figure 3 is a flow diagram of an additional embodiment of a method for modifying a value stored in a revision identification register. At block 310, a pre-operating system software agent is executed. At block 320, a value stored in a revision identification register is read. At block 330, a determination is made as to whether to modify the value stored in the revision identification register. If the determination is made to not modify the value stored in the revision identification register, then at 350 an operating system is loaded. If the determination is made to modify the value stored in the revision identification register, then the value is modified at block 340 before loading the operating system at block 350. The operations indicated at blocks 320, 330, and 340 are performed under control of the pre-operating system software agent.

[0025] In the foregoing specification the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than in a restrictive sense.

[0026] Reference in the specification to "an embodiment," "one embodiment," "some embodiments," or "other embodiments" means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least some embodiments, but not necessarily all embodiments, of the invention. The various

appearances of "an embodiment," "one embodiment," or "some embodiments" are not necessarily all referring to the same embodiments.

CLAIMS

What is claimed is:

1. An apparatus, comprising:

a revision identification register that allows modification of the revision identification register contents.

2. The apparatus of claim 1, further comprising:

a revision identification modification register, the revision identification register allowing modification of the revision identification register contents when indicated by the contents of the revision identification modification register.

3. The apparatus of claim 2, wherein the revision identification modification register includes a single bit, the state of the bit indicating whether the contents of the revision identification register are currently modifiable.

4. The apparatus of claim 3, wherein a value of "1" in the revision identification modification register indicates that the revision identification register will accept any value written to the revision identification register.

5. The apparatus of claim 4, wherein a value of "0" in the revision identification modification register indicates that the revision identification register will ignore any writes to the revision identification register.

6. A method, comprising:

determining whether a current revision identification value stored in a revision identification register indicates a first device stepping; and

replacing the current revision identification value with a revision identification value that indicates the first device stepping if the current revision identification value does not indicate the first device stepping.

7. The method of claim 6, further comprising:

ensuring that a revision identification modification register contains a value that indicates that the revision identification register will accept writes.

8. The method of claim 7, further comprising:

placing a value in the revision identification modification register that indicates that the revision identification register will not accept writes, the placing a value in the revision identification modification register occurring following replacing the current revision identification value with a revision identification value that indicates the first device stepping.

9. A method, comprising:

executing a pre-operating system software agent, the pre-operating software agent to determine whether to modify a value stored in a revision identification register;

modifying the value stored in the revision identification register; and

loading an operating system.

10. The method of claim 9, wherein determining whether to modify the value stored in the revision identification register includes determining whether the value stored in the revision identification register indicates a first device stepping.

11. The method of claim 10, wherein modifying the value stored in the revision identification register includes replacing the value stored in the revision identification register with a value that indicates the first device stepping if the value stored in the revision identification register does not indicate the first device stepping.

12. A machine-readable medium having stored thereon instructions which, when executed by a computer system, causes the computer system to perform a method comprising:

determining whether a current revision identification value stored in a revision identification register indicates a first device stepping; and

replacing the current revision identification value with a revision identification value that indicates the first device stepping if the current revision identification value does not indicate the first device stepping.

13. The machine readable medium of claim 12 having stored thereon additional instructions which, when executed, perform:

ensuring that a revision identification modification register contains a value that indicates that the revision identification register will accept writes.

14. The machine readable medium of claim 13 having stored thereon additional instructions which, when executed, perform:

placing a value in the revision identification modification register that indicates that the revision identification register will not accept writes, the placing a value in the revision identification modification register occurring following replacing the current revision identification value with a revision identification value that indicates the first device stepping.

15. A machine-readable medium having stored thereon instructions which, when executed by a computer system, causes the computer system to perform a method comprising:

executing a pre-operating system software agent, the pre-operating software agent to determine whether to modify a value stored in a revision identification register;

modifying the value stored in the revision identification register; and

loading an operating system.

16. The machine readable medium of claim 15, wherein determining whether to modify the value stored in the revision identification register includes determining whether the value stored in the revision identification register indicates a first device stepping.

17. The machine readable medium of claim 16, wherein modifying the value stored in the revision identification register includes replacing the value stored in the revision identification register with a value that indicates the first device stepping if the value stored in the revision identification register does not indicate the first device stepping.

18. A system, comprising:

a processor; and

a system logic device couple to the processor, the system logic device including

a revision identification register that allows modification of the revision identification register contents.

19. The system of claim 18, wherein the system logic device further includes a revision identification modification register, the revision identification register allowing modification of the revision identification register contents when indicated by the contents of the revision identification modification register.

20. The system of claim 19, wherein the revision identification modification register includes a single bit, the state of the bit indicating whether the contents of the revision identification register are currently modifiable.

21. The system of claim 20, wherein a value of "1" in the revision identification modification register indicates that the revision identification register will accept any value written to the revision identification register.

22. The system of claim 21, wherein a value of "0" in the revision identification modification register indicates that the revision identification register will ignore any writes to the revision identification register.

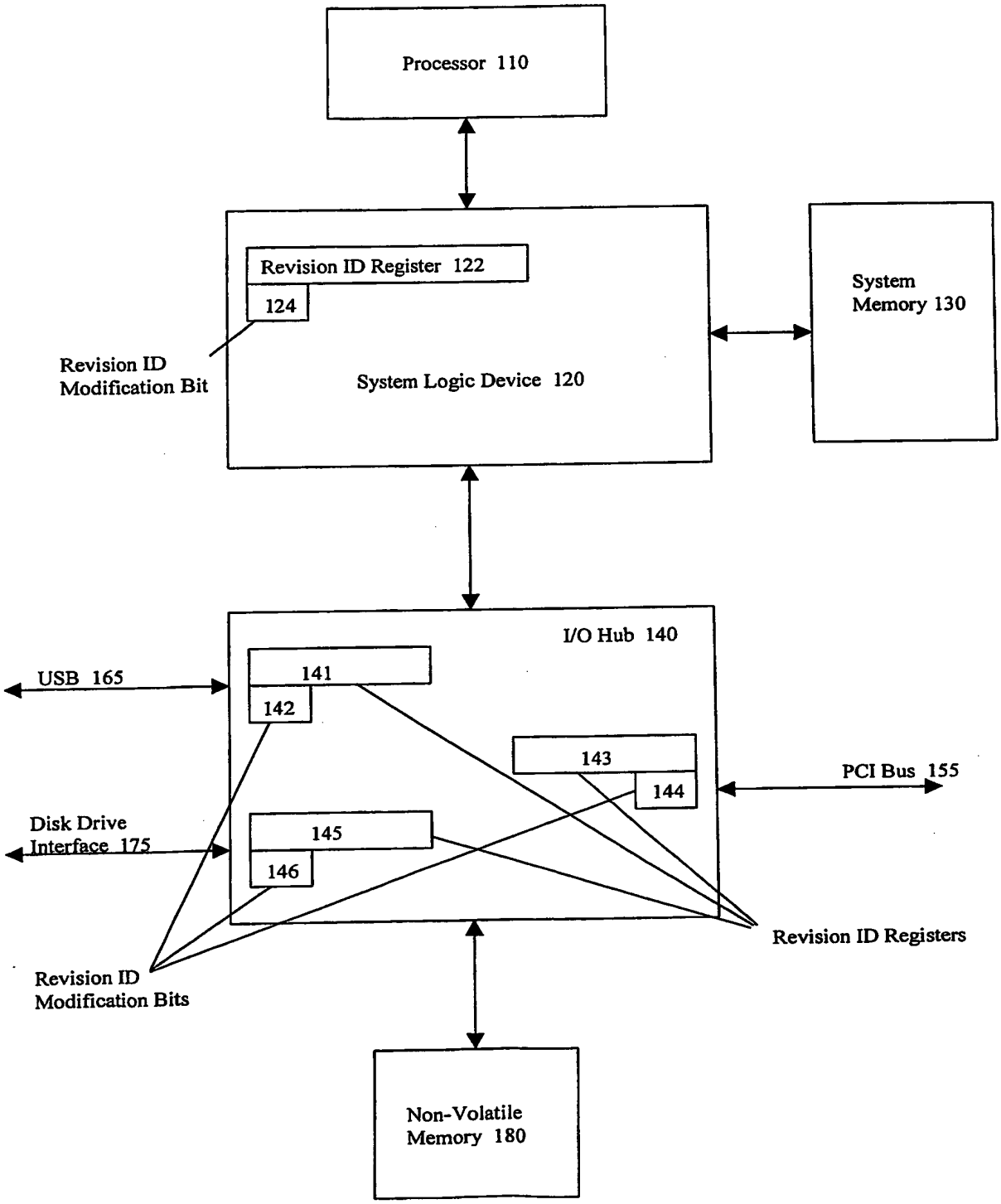

Processor 110
Revision ID Register 122
124
Revision ID
Modification Bit
System Logic Device 120
System
Memory 130
I/O Hub 140
USB 165
141
142
143
144
PCI Bus 155
Disk Drive
Interface 175
145
146
Revision ID Registers
Revision ID
Modification Bits
Non-Volatile
Memory 180

Figure 1

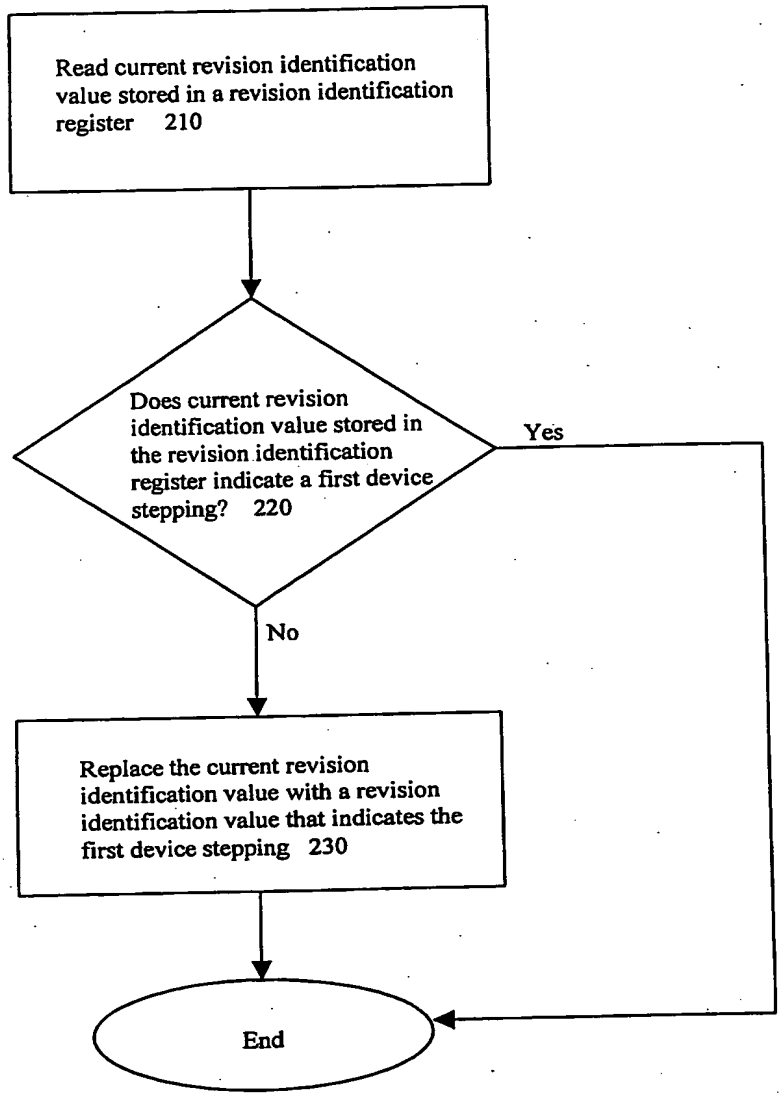

Read current revision identification value stored in a revision identification register 210
Does current revision identification value stored in the revision identification register indicate a first device stepping? 220
Yes
No
Replace the current revision identification value with a revision identification value that indicates the first device stepping 230
End

Figure 2

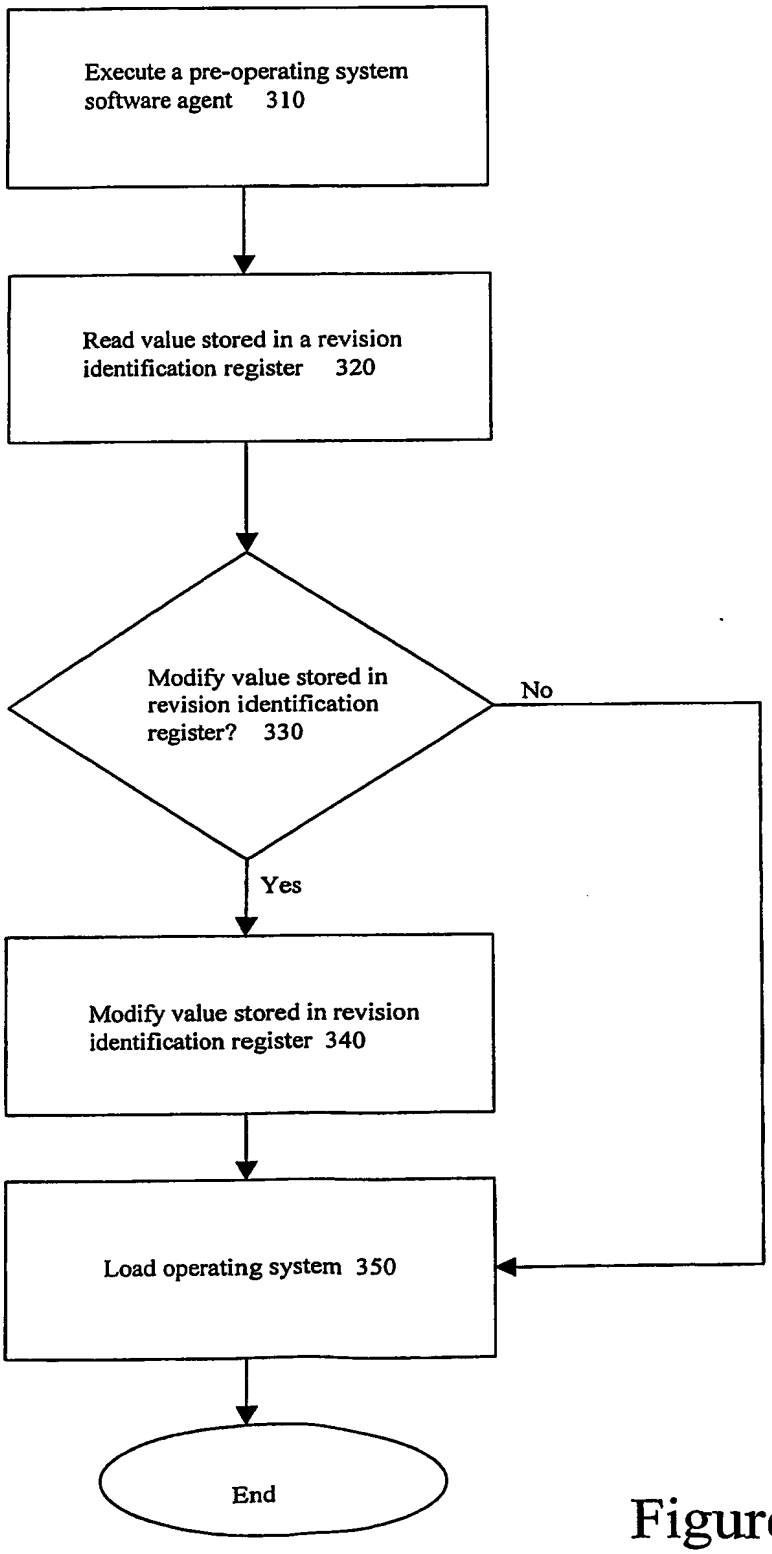

Execute a pre-operating system software agent 310
Read value stored in a revision identification register 320
Modify value stored in revision identification register? 330
No
Yes
Modify value stored in revision identification register 340
Load operating system 350
End

Figure 3

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 July 2003 (17.07.2003)

PCT

(10) International Publication Number
WO 03/058439 A3

(51) International Patent Classification[7]: G06F 13/38, 13/10

(21) International Application Number: PCT/US02/39159

(22) International Filing Date: 6 December 2002 (06.12.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/039,652 31 December 2001 (31.12.2001) US

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventors: **NALAWADI, Rajeev**; 690 Ahostini Circle, Folsom, CA 95630 (US). **SIDDIQI, Faraz**; 816 Canyon Terrace Lane, Folsom, CA 95630 (US).

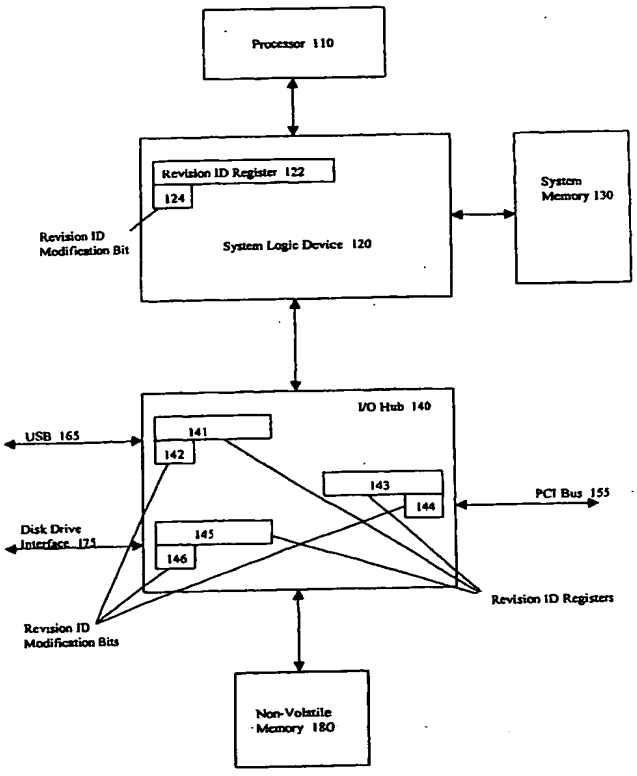
(74) Agent: **MALLIE, Michael, J.**; Blakely, Sokoloff, Taylor & Zafman, 7th Floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR MODIFYING THE CONTENTS OF A REVISION IDENTIFICATION REGISTER



(57) Abstract: An embodiment for modifying the contents of a revision identification register includes a revision identification register that is both readable and writable (the contents of the revision identification register are modifiable). A revision identification modification bit is also included. The contents of the revision identification register are only modifiable when the revision identification modification bit is set to indicate that writes to the revision identification register will be accepted.

WO 03/058439 A3

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

(88) Date of publication of the international search report:
6 November 2003

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/US 02/39159

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F13/38 G06F13/10

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
WPI Data, INSPEC, EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	US 5 826 075 A (BEALKOWSKI RICHARD ET AL) 20 October 1998 (1998-10-20) column 4, line 23 - line 39 column 9, line 40 -column 12, line 29 abstract ---	1,6,9, 12,15,18 2-5,7,8, 10,11, 13,14, 16,17, 19-22
A	US 5 737 585 A (KANESHIMA TOSHIHITO) 7 April 1998 (1998-04-07) column 2, line 11 - line 19 column 5, line 11 -column 6, line 52 abstract --- -/--	1-22

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 September 2003	29/09/2003

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Nguyen Xuan Hiep, C

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 02/39159

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 596 738 A (POPE STEVEN M) 21 January 1997 (1997-01-21) abstract; claims 1-3; figure 1	1-22
A	EP 0 472 433 A (FUJITSU LTD) 26 February 1992 (1992-02-26) column 1, line 48 -column 2, line 31 column 5, line 21 -column 6, line 11 abstract	1-22

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 02/39159

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5826075 A	20-10-1998	US 5878256 A	02-03-1999
		JP 3033642 B2	17-04-2000
		JP 5233272 A	10-09-1993
		JP 1998481 C	08-12-1995
		JP 5216654 A	27-08-1993
		JP 7034175 B	12-04-1995
US 5737585 A	07-04-1998	JP 7210395 A	11-08-1995
US 5596738 A	21-01-1997	NONE	
EP 0472433 A	26-02-1992	AU 630038 B2	15-10-1992
		AU 8268391 A	16-04-1992
		CA 2049690 A1	24-02-1992
		DE 69129480 D1	02-07-1998
		DE 69129480 T2	01-10-1998
		EP 0472433 A2	26-02-1992
		JP 2088488 C	02-09-1996
		JP 4357525 A	10-12-1992
		JP 8001594 B	10-01-1996
		US 5450589 A	12-09-1995